SCHEDULING OF COURSES AT A NAVAL TRAINING FACILITY

Floyd Tamerlane Samms

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SCHEDULING OF COURSES AT A
NAVAL TRAINING FACILITY

by

Floyd Tamerlane Samms, Jr.

Thesis Advisor: A. W. McMasters

September 1973

Scheduling of Courses

at a

Naval Training Facility

by

Floyd Tamerlane Samms, Jr.
Lieutenant, United States Navy
B.S., United States Naval Academy 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1973

ABSTRACT

An algorithm was developed to schedule courses at the Naval Guided Missiles School that would minimize the peak demand for a given single resource.

The algorithm will provide a schedule for any given number of courses of equal length. An extension to the algorithm will provide a schedule for any number of courses of unequal length.

The algorithm was developed in two phases. The first phase concerned finding an initial schedule. The second phase concerned reducing the maximum demand for the resource.

TABLE OF CONTENTS

I. INTRODUCTION

A. PROBLEM MOTIVATION

To function effectively, a weapons system as vast and complex as the Fleet Ballistic Missile system (FBM), requires competent and highly skilled technicians. The scheduling of the required training courses at a training facility is a vital part of ensuring the FBM system is manned with the necessary expertise to operate effectively.

The Office of the Chief of Naval Operations (OPNAV) determines the requirements for technicians to man the FBM force based upon policy objectives as promulgated by the Department of Defense. The Bureau of Naval Personnel (BUPERS) is charged with the responsibility of meeting the requirements specified by OPNAV. BUPERS devises the content of the several training curricula necessary to fulfill the technician requirements. The Naval Schools Commands are assigned the responsibility of conducting the training curricula. Training for a portion of the FBM personnel is conducted at the Naval Guided Missiles School, Dam Neck, Virginia. At Dam Neck, courses of instruction are available for commanding officers, executive officers, department heads, division officers and the following technical ratings: Fire Control Technician (FT); Electronics Technician (ET); Missile Technician (MT); and Torpedoman (TM).

The training cycle for FBM enlisted technicians consists of recruit training, "A" school, "B" school, submarine school, team and refresher training and operational experience. A man learns the basic skills of his particular rating in "A" school. The "B" school provides the advanced technical training related to his specific job in the FBM system. Team

and refresher training provide periodic review of basic knowledge and incorporate the latest technological advices of the FBM system. These last three areas of training are available at the Naval Guided Missiles School.

The FBM system is presently in transition from utilization of the POLARIS missile to the POSEIDON missile. The POSEIDON missile system while similar to POLARIS requires technician familiarization training. As a result, the Guided Missiles School is faced with the possibility of having to build an additional building to house the required classrooms and laboratories at a projected cost of $35.5 million.

Given the resource requirements, such as number of instructors, size of classrooms, size of the laboratories, for the convenings in the training cycle, a schedule to minimize the peak demands for these resources would keep the required number of instructors, classrooms and laboratories as small as possible and thereby keep the cost of the building as low as possible.

Burgess and Killebrew [1] present an heuristic approach to scheduling project activities by resource leveling. The objective they use is the minimization of the square of the resource requirement in each time segment. The solution seeks to keep a level demand for the resource throughout the scheduling cycle. If the conducting of a course by the training facility is defined as completing a project, then the school scheduling problem may be viewed as a multi-project scheduling problem.

Willingham [2] determined the optimum number of convenings for a series of courses in a one year time period utilizng three resources. He then presented an heuristic method to schedule the convenings into the one year period. He attempted to balance the schedule by keeping the number of convenings per course in session at any time during the period constant.

B. DEFINITION OF THE PROBLEM

Given the required student input into the FBM system as determined by higher authority and the required teaching resources as determined by the training facility, an optimal scheduling process is necessary to make economic use of the resources.

The development of an algorithm that will minimize the peak demands of given resources is required. The development of an algorithm that will minimize the peak demand for a single resource is the first step towards such an algorithm.

C. SCOPE OF THE STUDY

The problem will be approached by developing a scheduling algorithm for a single resource for courses of equal length and the first steps of a scheduling algorithm for courses of unequal length.

II. THE SCHEDULING ALGORITHM

A. PREVIEW

The central idea behind the algorithm methodology is to rearrange the initial schedule in order to reduce the maximum demand for the resource. The algorithm begins by scheduling the convenings in order relative to their demand for the resource; those with the greatest demand are scheduled first and those with the least demand last. The algorithm then attempts to interchange convenings scheduled into the period of maximum demand and convenings having a lesser demand for the resource scheduled into any other time period, thereby reducing the maximum demand.

The algorithm presented will provide a schedule for courses of equal length of which the time period T is an integer multiple.

Before stating the algorithm it is necessary to define some terms.

Let T = the length of the fixed scheduling period.

I = number of different courses which must be taught during T.

N_i= number of convenings of course i, $(i=1,2,\ldots,I)$.

t_i= the length of one convening of course i; $t_i \leq T$.

c_i= the integer resource demand coefficient of course i.

Clearly, if the combined lengths of all required convenings does not exceed T, that is,

$$\sum_i N_i t_i < T,$$

then the courses should be scheduled sequentially in the time zero to T, $[0,T]$. Thus the maximum demand for the resource will be the largest of the resource demand coefficients, c_i.

Assuming the problem is other than the trivial case stated above, a scheduling cycle of length T is defined as a fraction (<1.0) of a schedule. Hence a schedule will consist of at least two cycles.

Using a matrix representation for the problem, an mxn matrix $\left[o_{ij}\right]$ may be defined where o_{ij} denotes the resource demand coefficient, c_k, of the convening scheduled into the j^{th} segment of the i^{th} scheduling cycle.

The dimension of the matrix, mxn, is determined by the algorithm; n is equal to the number of time segments in the time period T, and m is twice the number of scheduling cycles needed to complete the schedule minus one.

The problem to be solved by the algorithm may be stated as, find that schedule of course convenings which minimizes

$$z = \max_{j} \sum_{i} o_{ij} ,$$

the largest resource demand to occur at any time during the period T.

B. THE ALGORITHM FOR COURSES OF EQUAL LENGTH

The algorithm begins by determining an initial schedule. An attempt is then made to decrease the maximum demand for the resource by rescheduling any combination of courses. The reschedule is first attempted between the period of maximum demand and the period of minimum demand. If this is not possible, the algorithm then tries to reschedule courses between the period of maximum demand and any other period.

After the initial schedule is obtained, the algorithm compares each single course scheduled into the period of maximum demand with each single course scheduled into the period of minimum demand in an attempt to find two courses whose interchange will reduce the maximum demand for the resource. If no single course can be found in the minimum period to be used in an interchange, the algorithm successively compares each single course

in the maximum period with all pairs, triplets, etc., of courses in the minimum period. If the total demand for these combinations of courses will not reduce the maximum demand when interchanged with a single course of the maximum period, the algorithm compares first pairs, then triplets and so on of courses in the maximum period with combinations of courses in the minimum period. If after all possible combinations of courses in the maximum period are compared with all combinations of courses in the minimum period and no interchange of courses can be made, the algorithm selects another time period in place of the minimum period and starts the process again. After all time periods have been investigated and fail to produce an interchange, the algorithm terminates.

1. Statement of the Algorithm

The number of time segments in the time period T can be calculated by

$$TS = \frac{T}{t}$$

where

TS = the number of time segments.

t = the length of one convening of any course.

1. There is a resource demand coefficient c_i for each convening of course i. Hence there are a total of $\sum N_i$ coefficients. Arrange these demand coefficients into a non-increasing sequence $\{s_j\}$ where $j = 1,2,\ldots,\sum N_i$. Compute P where

$$P = \frac{\sum N_i}{TS} \quad .$$

If $p = \frac{\sum N_i}{TS}$ is an integer, then add (P-1)TS zero elements to the end of the sequence. If P is not an integer, then add (K-P)TS + (K-1)TS=(2K-P-1)TS

zero elements to the end of the sequence where K is the smallest integer larger than P. Schedule the first cycle of convenings by placing $s_1, s_2, \ldots, s_{TS}$ into the $(m,1), (m,2), \ldots, (m,TS)$ positions; e.g., the bottom row of the matrix. The second cycle is obtained by scheduling backwards over the period $[0,T]$ and adding the $(m-1)^{st}$ row to the matrix. Thus, this row consists of $s_{2TS}, s_{2TS-1}, \ldots, s_{TS+1}$. The schedule and the matrix are completed by continuing to schedule forward on the odd numbered cycles and backwards on the even numbered cycles until the sequence $\{s_j\}$ is exhausted.

2. Compute the value of δ where

$$\delta = \max_j \sum_i o_{ij} - \min_j \sum_i o_{ij}.$$

If $\delta = 0$ or 1, then stop; the solution is optimal. Otherwise go on to step 3.

3. Designate the columns of maximum and minimum sums from the matrix as $\bar{o}_u$ and $\bar{o}_v$, respectively. In case of ties, select a column at random. Let o_u and o_v represent some single element of $\bar{o}_u$ and $\bar{o}_v$. Set $x = o_{mu}$ and go to step 4.

4. Set $y = \max \{o_{iv} < x\}$. Interchange the elements of $\bar{o}_u$ and $\bar{o}_v$ corresponding to x and y if $x - \delta < y < x$ and go to step 10. Otherwise, reduce x to the next largest element value in $\bar{o}_u$ and

   a. if $x > 0$, repeat step 4.

   b. if $x = 0$, set $x = o_{mu}$, $y_1 = \max \{o_{iv} < x\}$ and go to step 5.

5. Denote the element corresponding to y_1 as o_{pv}.

   Set $y_2 = \max \{o_{iv} \mid i < p,\ 0 < o_{iv} < x - y_1\}$.

   a. If an element of $\bar{o}_v$ can be found to use as y_2 then go to step 6.

b. If no element of $\overline{o}_v$ can be found to use as y_2 then reduce y_1 to the value of the element in $\overline{o}_v$ which is the next smaller than its current value and again try to find a y_2 value. If no y_2 value is found even when y_1 is reduced to the second smallest non-zero element in $\overline{o}_v$, go to step 9.

6. Interchange the elements of $\overline{o}_v$ corresponding to y_1 and y_2 with the element of $\overline{o}_u$ corresponding to x and any zero element of $\overline{o}_u$ if $x - \delta < y_1 + y_2 < x$ and go to step 10. Otherwise, go to step 7.

7. Denote the element corresponding to y_2 as o_{qv}.

Set $y_3 = \max \left\{ o_{iv} \;\middle|\; i < q \,,\, 0 < o_{iv} < x - y_1 - y_2 \right\}$.

a. If an element of $\overline{o}_v$ can be found to use as y_3 then set k = 3 and go to step 8.

b. Otherwise, reduce y_2 to the value of the next smaller positive $\overline{o}_v$ element if one exists and return to step 6. If no positive element exists to use as y_2 return to step 5b.

8. Continue the process begun in steps 6 and 7 of trying to interchange groups of elements corresponding to $y_1, y_2, \ldots, y_k$ from $\overline{o}_v$ with the element corresponding to x and k-1 zero elements in $\overline{o}_u$. For example, search next for a y_4 satisfying $y_4 = \max \left\{ o_{iv} \;\middle|\; i < r, o < o_{iv} < x - y_i - y_2 - y_3 \right\}$. If no y_4 exists then reduce y_3 and again attempt an interchange with y_1, y_2, and y_3 if $y_3 > 0$. If $y_3 = 0$, go to step 7b.

9. Decrease x to the value of the next largest element of $\overline{o}_u$. If $x > 0$, set $y_1 = \max \left\{ o_{iv} < x \right\}$ and return to step 5. Otherwise, set $x = o_{mu} + o_{m-1,u}$ and go to step 11.

10. Rearrange the elements of $\overline{o}_u$ and $\overline{o}_v$ into a non-increasing sequence beginning with the bottom element of each column. Go to step 2.

11. Set $y = \max \left\{ o_{iv} < x \right\}$. Interchange the elements of $\overline{o}_u$ and $\overline{o}_v$ corresponding to the elements used in the x and y values if $x - \delta < y < x$ and go on to step 10. Otherwise, reduce x as follows:

$$\text{current x value} = o_{pu} + o_{p-1,u} ;$$

$$\text{new x value} = o_{p-1,u} + o_{p-2,u}.$$

a. If $x > 0$, repeat step 11.

b. if $x = 0$, set $x = o_{mu} + o_{m-1,u}$ and proceed as in steps 5 through 9 in an attempt to exchange a pair of elements of $\overline{o}_u$ with two or more elements from $\overline{o}_v$. If no interchanges are possible go to step 12.

12. Successively increase the number of elements in $\overline{o}_u$ used in the computation of x by one more element and continue the process of first comparing these elements with single elements of $\overline{o}_v$ and then with pairs, triplets, etc. If no interchange between $\overline{o}_u$ and $\overline{o}_v$ are possible then define $\overline{o}_v$ as the column having the next smallest sum and return to step 2.

2. Example 1

This example will demonstrate the forward and backwards scheduling method for obtaining an initial schedule.

Let us suppose a training facility must offer five courses during a ten-month training cycle according to the following table:

Table 1. Problem Description for the Example of Case I

Training Course i	N_i	t_i	c_i
1	6	1	2
2	9	1	9
3	1	1	8
4	8	1	3
5	10	1	4

where

N_i = the required number of convenings of course i.

t_i = the length in months of one convening of course i.

c_i = the required laboratory space, in square feet, for one convening of course i.

The resource demand coefficients c_i are arranged into a non-increasing sequence: $\{9,9,9,9,9,9,9,9,9,8,4,4,4,4,4,4,4,4,4,4,3,3,3,3,3,3,3,3,2,2,2,2,2,2,$ $0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$. The total time period, T, is ten months. TS is determined to be 10. Therefore the segment length must be one month. The matrix will be a mx10 matrix. The first scheduling cycle is completed by scheduling the first ten elements of the sequence.

$$\begin{pmatrix} \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8 \end{pmatrix}$$

The second cycle is completed by scheduling the next ten elements of the sequence backwards over the period.

$$\begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8 \end{pmatrix}$$

The third cycle is completed by forward scheduling elements twenty-one through thirty.

$$\begin{pmatrix} \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8 \end{pmatrix}$$

The fourth cycle is completed by scheduling the next ten elements.

$$\begin{array}{cc} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8 \end{pmatrix} \\ \hline \sum_i o_{ij} = & \begin{matrix} 16 & 16 & 16 & 16 & 16 & 16 & 18 & 18 & 17 & 16 \end{matrix} \end{array}$$

The matrix is completed by adding the zero elements.

$$\begin{array}{cc} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8 \end{pmatrix} \\ \hline \sum_i o_{ij} = & \begin{matrix} 16 & 16 & 16 & 16 & 16 & 16 & 18 & 18 & 17 & 16 \end{matrix} \end{array}$$

Column seven is designated $\bar{o}_u$. Column six is designated $\bar{o}_v$; $\delta = 18 - 16 = 2$. The value for $x = o_{mu}$ is 9 as required by step 4 of the algorithm. For $x = 9$, a value for y satisfying $x - \delta < y < x$ cannot be found.

The value of x is reduced to 4, the next largest element in $\bar{o}_u$, as specified by step 4 and the search for y is repeated. A value of $y = 3$ satisfies the criterion. Interchanging the elements corresponding to x and y, rearranging the columns required by step 10 not being necessary, results in the schedule:

$$
\begin{array}{r}
\\ \\ \\ \\ \\ \\ \\
\sum_i o_{ij} =
\end{array}
\begin{array}{c}
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 2 & 2 \\
4 & 4 & 4 & 4 & 4 & 4 & 3 & 4 & 4 & 4 \\
9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8
\end{pmatrix} \\ \hline
16 \;\; 16 \;\; 16 \;\; 16 \;\; 16 \;\; 17 \;\; 17 \;\; 18 \;\; 17 \;\; 16
\end{array}
$$

Returning to step 2, column eight is $\overline{o}_u$, column ten is $\overline{o}_v$ and $\delta = 18 - 16 = 2$. Proceeding as before in the search for an acceptable value for y, step 4 is repeated until x has been reduced to 3. A value for y = 2 qualifies to be interchanged with x. Rearranging the matrix as required by step 10 produces the schedule:

$$
\begin{array}{r}
\\ \\ \\ \\ \\ \\ \\
\sum_i o_{ij} =
\end{array}
\begin{array}{c}
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 2 & 3 \\
4 & 4 & 4 & 4 & 4 & 4 & 3 & 4 & 4 & 4 \\
9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8
\end{pmatrix} \\ \hline
16 \;\; 16 \;\; 16 \;\; 16 \;\; 16 \;\; 17 \;\; 17 \;\; 17 \;\; 17 \;\; 17
\end{array}
$$

The schedule is optimal because $\delta = 17 - 16 = 1$.

3. Example 2

Let us suppose the initial schedule for a particular problem is given below.

$$
\begin{array}{r}
\\ \\ \\ \\ \\ \\ \\
\sum_i o_{ij} =
\end{array}
\begin{array}{c}
\begin{pmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
2 & 3 \\
5 & 3 \\
7 & 9 \\
13 & 10
\end{pmatrix} \\ \hline
27 \;\; 25
\end{array}
$$

Column one is designated $\overline{o}_u$. Column two is designated $\overline{o}_v$; $\delta = 27 - 25 = 2$. The value for $x = 0_{mu}$ is 13 as required by step 4 of the algorithm. For

$x = 13$, a value for y of step 4 such that $x - \delta < y < x$ cannot be found.

The value for x is reduced to 7, the next largest element in $\overline{o}_u$. Again no value for y will satisfy the criterion $x - \delta < y < x$. This is also true for $x = 5$ and $x = 2$. The value for x is reduced to zero and the algorithm looks for a pair of elements in $\overline{o}_v$ to interchange with x.

The value for x is again set to $o_{mu} = 13$. The largest element of $\overline{o}_v$ less than x, designated y_1, is 10. The second element of the pair, y_2, must satisfy $10 + y_2 < 13$ with $y_2 > 0$. No such y_2 can be found. Therefore, the value of y_1 is reduced to 9 as specified by step 5 and the search for y_2 is repeated. This time a value of 3 is found to use as y_2. The pair (9,3) from the second column can therefore be interchanged with the element 13 and any of the 0 elements in the first column. Rearrangement of the elements according to step 10 results in the following:

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 2 & 0 \\ 3 & 0 \\ 5 & 3 \\ 7 & 10 \\ 9 & 13 \end{pmatrix}$$

$$\sum_i o_{ij} = \quad 26 \quad 26$$

Since $\delta = 0$ upon returning to step 2, the solution is optimal.

C. THE ALGORITHM EXTENDED TO COURSES OF UNEQUAL LENGTH

When the scheduling problem is extended to courses of unequal length, it becomes more difficult. Each convening will now span one or more time segments depending upon the individual course length. It can no longer be guaranteed that a scheduling cycle will be complete. There may be gaps, due to the unequal course lengths, where no convening may be scheduled.

The solution method becomes more complicated in that interchanging of convenings to reduce the maximum demand for the resource must make allowance for the unequal course lengths. The algorithm must be modified such that reducing the maximum demand will consist of a series of one or more convenings being interchanged with another series of convenings spanning the same number of time segments.

The algorithm presented in this section considers only interchanging individual convenings. Clearly, interchanging series of convenings instead of individual convenings becomes especially difficult when the algorithm must consider pairs, triplets, etc., of convenings to interchange and will be left as the subject of a future study.

1. Statement of the Algorithm

The number of time segments in the time period T can be calculated by redefining the variables TS and t.

Let $$TS = \frac{T}{t}$$

where

TS = the number of time segments

t = the greatest common divisor for all course lengths and T.

Due to the unequal lengths of the courses, some convenings may span more than one time segment. The number of time segments spanned by each convening is the ratio

$$TS_i = \frac{t_i}{t}$$

where

TS_i = number of time segments spanned by one convening of course i

t_i = length of one convening of course i.

1. There is a demand coefficient for each convening of course i. Each convening may be represented by a group of identical coefficients indicating the number of time segments spanned by the convening; e.g., (c_i, c_i, c_i) if course i spans three segments. There are $\sum N_i$ coefficient groups. A convening of only one time segment in length will be represented by a single integer without parentheses. Arrange these demand coefficients into a non-increasing sequence of groups $\left\{ (s^1_j, s^1_j, \ldots, s_j^{TS_j}) \right\}$ where $j = 1,2,\ldots,\sum N_i$. Begin scheduling the first cycle of convenings by placing $s_1^1, s_1^2, \ldots, s_1^{TS_j})$ into the $(m,1),(m,2),\ldots,(m,TS_1)$ positions of the matrix. Complete the first cycle by placing groups into the (m,TS_1+1), $(m,TS_1+2),\ldots,$ (m,TS) position until the (m,TS) position is filled or a complete group cannot be scheduled into the time remaining due to its length. In this latter situation the bottom matrix row is completed with zero-element groups. The (m-1) row is then scheduled using the backwards pass and adding zeros when needed to fill up the row. The schedule and the matrix are completed by continuing to schedule forward on the odd-numbered cycles and backwards on the even-numbered cycles until the sequence $\left\{ (s_j^1, s_j^2, \ldots, s_j^{TS_j}) \right\}$ is exhausted.

2. Compute the value of δ where

$$\delta = \max_j \sum_i o_{ij} - \min_j \sum_i o_{ij}.$$

If $\delta = 0$ or 1, then stop; the solution is optimal. Otherwise go to step 3.

3. Designate the column of maximum sum from the matrix as $\bar{o}_u$. In case of a tie, select a column at random. Let o_u represent some single element of $\bar{o}_u$. Define the group containing o_u as $(o_u^1, o_u^2, \ldots, o_u^{TS_w})$ where w is the position number of the course for which o_u is the resource demand coefficient in the demand coefficient sequence of step 1. Let o_v

represent any other single element of the matrix not contained in $\overline{o}_u$. Define the group containing o_v as $(o_v^1, o_v^2, \ldots, o_v^{TS_y})$ where y is the position number of the course for which o_v is the resource demand coefficient in the demand coefficient sequence. Let $\left\{(s_k^1, s_k^2, \ldots, s_k^{TS_k})\right\}_{ij\lambda}$ be a subsequence of P demand coefficient groups starting with the (i,j) position of the matrix and spanning $\lambda \leq TS$ time segments with $\lambda = \sum_{k=1}^{P} TS_k$. Each subsequence will consist of a portion (≤ 1) of row i. Set o_u = the maximum valued non-zero integer element in $\overline{o}_u$.

Search the matrix for an element o_v which satisfies the following conditions:

i. $o_u - \alpha < o_v < o_u$ where $\alpha = \sum_i o_{iu} - \sum_i o_{iv}$

ii. $(o_v^1, o_v^2, \ldots, o_v^{TS_y}) \in \left\{(s_k^1, s_k^2, \ldots, s_k^{TS_k})\right\}_{ij\lambda}$ for some i,j, λ having $\lambda = TS_w$

iii. $(s_r^1, s_r^2, \ldots, s_r^{TS_r}) \in \left\{(s_k^1, s_k^2, \ldots, s_k^{TS_k})\right\}_{ij\lambda}$ has $s_r \leq 0_u$ for all r.

If such an element can be found, interchange the group $(o_u^1, o_u^2, \ldots, o_u^{TS_w})$ with the subsequence $\left\{(s_k^1, s_k^2, \ldots, s_k^{TS_k})\right\}_{ij\lambda}$ in their respective columns and return to step 2. If no interchange is possible set o_u equal to the next largest integer in $\overline{o}_u$ and repeat the search. If no interchange is possible for o_u being any non-zero element of $\overline{o}_u$, then terminate.

2. Example 1

Let us suppose the training facility must now offer courses of various lengths as given in Table II. Again it is desired to minimize the peak demand for the resource.

Table II: Problem Description for the Example of Case II

Training Course i	N_i	t_i	c_i
1	4	1	1
2	6	3	2
3	4	4	3
4	8	2	4
5	4	1	6

where

c_i = the number of classrooms required for the student enrollement of course i.

The training cycle, or time period T, is specified as one year. The greatest common divisor of all course lengths and the time period is one month. Therefore TS is equal to 12 time segments. The values for TS_i can be calculated as given in Table III.

Table III: Time Segments per Course for the Example of Case II

Training Course i	TS_i
1	1
2	3
3	4
4	2
5	1

The sequence of resource demand coefficients is: $\{6,6,6,6,(4\ 4),(4\ 4)$ $(4\ 4),(4\ 4),\ (4\ 4),\ (4\ 4),\ (4\ 4),\ (4\ 4),\ (3\ 3\ 3\ 3),\ (3\ 3\ 3\ 3),\ (3\ 3\ 3\ 3),$ $(3\ 3\ 3\ 3),\ (2\ 2\ 2),\ (2\ 2\ 2),\ (2\ 2\ 2),\ (2\ 2\ 2),\ (2\ 2\ 2),\ (2\ 2\ 2),\ 1,1,1,1\}$.

There are twelve time segments in the schedule, therefore the segment length is one month. The matrix will be an mx12 matrix.

The initial solution is obtained using the forward and backward method discussed in Case I.

$$
\begin{array}{c}
\begin{pmatrix}
(2 & 2 & 2) & (2 & 2 & 2) & 1 & 1 & 1 & 1 & 0 & 0 \\
(2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) \\
(3 & 3 & 3 & 3) & (3 & 3 & 3 & 3) & (3 & 3 & 3 & 3) \\
(3 & 3 & 3 & 3) & (4 & 4) & (4 & 4) & (4 & 4) & (4 & 4) \\
6 & 6 & 6 & 6 & (4 & 4) & (4 & 4) & (4 & 4) & (4 & 4)
\end{pmatrix} \\
\hline
\end{array}
$$

$$\sum_i o_{ij} = \quad 16\ 16\ 16\ 16\ 15\ 15\ 14\ 14\ 14\ 14\ 13\ 13$$

From step 2 of the algorithm we see $\delta = \max_j \sum_i o_{ij} - \min_j \sum_i o_{ij} = 16-13 = 3$. Therefore some rearrangement of the matrix may be necessary. Let column one be designated as the maximum column and set $o_u = 6$, the largest element of column one. For $o_u = 6$, condition i of step 4 of the algorithm is violated for all possible values of o_v. For $o_u = 3$, $o_v = 2$ will satisfy condition i. However, condition ii is violated. Reducing o_u to 2 and setting $o_v = 1$ produce $\alpha = 2$ and the conditions are satisfied. The group (2 2 2) and sub-sequence $\{1,1,1\}$ are interchanged.

$$
\begin{array}{c}
\begin{pmatrix}
(2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) & 1 & 0 & 0 \\
1 & 1 & 1 & (2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) \\
(3 & 3 & 3 & 3) & (3 & 3 & 3 & 3) & (3 & 3 & 3 & 3) \\
(3 & 3 & 3 & 3) & (4 & 4) & (4 & 4) & (4 & 4) & (4 & 4) \\
6 & 6 & 6 & 6 & (4 & 4) & (4 & 4) & (4 & 4) & (4 & 4)
\end{pmatrix} \\
\hline
\end{array}
$$

$$\sum_i o_{ij} = \quad 15\ 15\ 15\ 16\ 15\ 15\ 15\ 15\ 15\ 14\ 13\ 13$$

Returning to step 2, δ is again computed to be 16-13 = 3. Column four is now the maximum column. Set $o_u = 6$, the largest element of column four, and search the matrix for values of o_v. As before, an acceptable value for

o_v cannot be found until o_u has been reduced to 2. Setting $o_v = 1$, the group (2 2 2) is interchanged with the sub-sequence $\{1,0,0\}$.

$$
\begin{pmatrix}
(2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) \\
1 & 1 & 1 & 1 & 0 & 0 & (2 & 2 & 2) & (2 & 2 & 2) \\
(3 & 3 & 3 & 3)(3 & 3 & 3 & 3)(3 & 3 & 3 & 3) \\
(3 & 3 & 3 & 3)(4 & 4) & (4 & 4)(4 & 4)(4 & 4) \\
6 & 6 & 6 & 6\ (4 & 4) & (4 & 4)(4 & 4)(4 & 4)
\end{pmatrix}
$$

$$
\sum_i o_{ij} = \begin{array}{cccccccccccc} 15 & 15 & 15 & 15 & 13 & 13 & 15 & 15 & 15 & 15 & 15 & 15 \end{array}
$$

Trying to reduce the maximum value of 15 in column one and two results in two interchanges when o_u equals 1.

$$
\begin{array}{cccccccccccc}
(2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) & (2 & 2 & 2) \\
0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 \\
(3 & 3 & 3 & 3)(3 & 3 & 3 & 3)(3 & 3 & 3 & 3) \\
(3 & 3 & 3 & 3)(4 & 4) & (4 & 4)(4 & 4)(4 & 4) \\
6 & 6 & 6 & 6\ (4 & 4) & (4 & 4)(4 & 4)(4 & 4) \\
\hline
\end{array}
$$

$$
\sum_i o_{ij} = \begin{array}{cccccccccccc} 14 & 14 & 15 & 15 & 14 & 14 & 15 & 15 & 15 & 15 & 15 & 15 \end{array}
$$

The solution is optimal because $\delta = 15-14 = 1$.

It is interesting to note that the last two interchanges are not necessary to ensure optimality. Because the number of columns (10) having the maximum sum of 15 is greater than the number of columns (2) having the minimum sum of 13 and it would never be possible to reduce the max $\sum_i o_{ij}$ below 15.

D. COMMENTS ON OPTIMALITY

1. <u>The Algorithm</u>

There are three conditions under which the algorithm will terminate. The first two, $\delta = 0$ or $\delta = 1$, will produce an optimal schedule. There is no guarantee of optimality if the algorithm terminates with $\delta > 1$.

If $\delta = 0$, then the maximum column sum and the minimum column sum are equal. Clearly all column sums are equal and the z value is constant over the time period T. Therefore the interchanging of elements between time segments cannot reduce this value and the solution is optimal.

If $\delta = 1$, then the maximum column sum and the minimum column sum differ by only one unit. Since the elements of the matrix are non-negative integers, interchanging elements between the maximum and minimum columns will have one of three effects. The value of z will not change if the elements are equal. If the elements differ by one unit and $o_v < o_u$, the maximum and minimum columns will become the minimum and maximum columns, respectively. If they differ by one unit and $o_v > o_u$ then the maximum value would increase. The maximum value will also increase if they differ by more than one unit. In no case will z be reduced and the solution is optimal.

For $\delta > 1$, the maximum column and the column sums of all other columns differ by an amount less than the combined difference between any combination of elements in $\bar{o}_u$ and any other column. It is possible that some intermediate interchange between two $\bar{o}_v$ columns may allow a subsequent interchange to take place between some $\bar{o}_v$ and the current $\bar{o}_u$ column. Therefore no general guarantee of optimality can be provided.

2. The Algorithm Extended

The algorithm terminates under one of three conditions. If δ = 0 or 1, the previous discussion on optimality is germane.

If $\delta > 1$, the algorithm attempts to reschedule any group of elements in the maximum column that will reduce the z value. There is no provision in the algorithm to reschedule combinations of groups. For this reason there can be no guarantee of optimality.

III. DISCUSSION AND EXTENSIONS

A. DISCUSSION OF THE ALGORITHM

To obtain an initial solution to the algorithm and the extension, the sequence of convenings is placed into the matrix by scheduling forward on the odd-numbered cycles and backwards on the even-numbered cycles. While this is not the only way to initially complete the matrix, it is designated to accommodate the user. Because the algorithm is not dependent upon the method of obtaining an initial solution, the forward and backwards scheduling seems to be an easy way to get started.

B. ADDITIONAL OPTIMALITY CONDITIONS

There appear to be tests other than $\delta = 0$ or 1 for determining, at any time during the solution process, if optimality has been obtained. One such test involves problems having multiple columns of maximum sum and multiple columns of minimum sum. For example, in the case of $\delta = 2$, if the number of multiple columns of maximum sum is greater than the number of multiple columns of minimum sum, the maximum demand cannot be reduced and the solution is immediately optimal. For $\delta \geq 3$, the relationship of the number of multiple columns of maximum sum to the number of multiple columns of minimum sum, for which the solution is immediately optimal, is not easily determined.

Further study of optimality conditions should also involve interchanges between columns other than the maximum and minimum columns. Such a study may require an analysis of a large number of randomly generated examples. An analysis of this type would also be beneficial in appraising how well the algorithm obtains optimal solutions.

C. EXTENSION TO MORE THAN ONE RESOURCE

The scheduling problem where more than one resource is involved in complicated by the fact that minimization of each ignoring the others will undoubtedly result in completely different schedules, any of which does not minimize all resources. One approach would be to create a combined resource by weighting the individual resources according to their importance and adding them together. The algorithm described in this thesis could then be used to find a good schedule.

The more common problem is to minimize the demands for one unrestricted resource subject to restrictions on the maximum amount to resources available for the remainder of the resources. The preceding algorithms would have to be modified to generate an initial feasible schedule and then to insure that any column interchanges which take place do not violate feasibility. The development of such modifications should begin using the first model (all courses of equal length) since it provides the simplest structure for checking feasibility of proposed interchanges.

BIBLIOGRAPHY

1. Burgess, A.R. and Killebrew, J.B., "Variation in Activity Level on a Cyclical Arrow Diagram," Journal of Industrial Engineering, v. 13, March-April 1962.

2. Willingham, G.W., Determining the Optimal Flow of Student Sections at the Naval Guided Missiles School, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor Alan W. McMasters, Code 55Mg Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	5
4.	Professor Gilbert T. Howard, Code 55 Hk Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	Chief of Naval Personnel Pers-11b Department of the Navy Washington, D.C. 20390	1
7.	LT Floyd T. Samms, Jr., USN 8524 Benjamin Avenue Norfolk, Virginia 23518	1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SCHEDULING OF COURSES AT A NAVAL TRAINING FACILITY		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1973
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Floyd Tamerlane Samms, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California		12. REPORT DATE September 1973
		13. NUMBER OF PAGES 29
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Project Scheduling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm was developed to schedule courses at the Naval Guided Missiles School that would minimize the peak demand for a given single resource. The algorithm will provide a schedlue for any number of courses of equal length. An extension to the algorithm will provide a schedule for any number of courses of unequal length.
The algorithm was developed in two phases. The first phase concerned finding an initial schedule. The second phase concerned reducing the maximum demand for the resource.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD Form 1473 (BACK)
1 Jan 73
S/N 0102-014-6601

Thesis 146275
S1554 Samms
c.1
Scheduling of courses at a Naval training facility.